REMARKS

This is a full and timely response to the final Office Action mailed on April 2, 2008 (Paper No./Date 20080317). Upon entry of the foregoing amendments, claims 1-6 and 11-26 remain pending. Applicants reserve the right to pursue the subject matter of these canceled claims in a continuing application, if Applicants so choose, and do not intend to dedicate any of the canceled subject matter to the public. Reconsideration and allowance of the Application and present claims are respectfully requested.

Response to Claim Rejection Under 35 U.S.C. § 103

Claims 1-26 stand rejected under 35 U.S.C. §103 as allegedly being unpatentable over U.S. Patent No. 6,347,307 to *Sandhu, et al.* in view of U.S. Publication No. 2004/0083453 to *Knight*. Applicants respectfully traverse this rejection.

A. Limitation of XSLT

Extensible stylesheet language transformation (XSLT) is a generic transformation approach to work with many types of extensible markup language (XML) data ***for the transformation of XML documents into other XML documents***. XSLT is based on XML path language (Xpath) queries. During XSLT transformation, transformation layer typically goes through the following stages:

a. load the XML documents (source, style sheet) into memory and create source and style sheet trees.

b. apply the transformation process that looks like a composition of complex JOINS on this in-memory XML document with potentially hundreds of nested Xpath queries to product the result tree.

c. create the result tree in-memory.

d. serialize the in-memory result tree into XML document

The above-mentioned stages slow the performance of XSLT transformation. The claimed invention is developed to overcome this drawback. As mentioned in the Background of the application, "XSLT and other existing methods have not fully addressed the issues relating to data transformation in a communications network environment. According, there remains a need for a method and system device that solves existing shortcoming relating to data transformation." One way to increase performance is by transforming a first object type into a second object type rather than in XML format which is the result of an XSLT transformation. The second object type can be used in application programming interface and does not involve run-time interpretation of transformation specification as in XSLT transformation. This can increase the performance of the data transformation.

B. *Sandhu* and *Knight*

In *Sandhu*, data transformation is used to achieve end to end solution using XSLT. A portion of the Abstract of *Sandhu* recites:

> "Such communications are facilitated using a novel ***XML-based syntax (FinXML.TM.) and XSL-based processing language (FinScript.TM.)***. FinXML provides a standard data interchange language for capital market transactions and supports a broad set of elements and attributes that represent a wide variety of financial transactions, reference data, and market data. The common description of the FinXML syntax can be used for all aspects of straight-through-processing, including deal creation, confirmation, settlement, payment, risk management, and accounting."
>
> (Emphasis Added)

In *Knight*, XSLT is used to achieve data transformation. In fact, the Abstract of *Knight* recites:

> "The data provider may employ an external transform (e.g. an ***XSLT***). Widgets receive logical, presentation-centric data instances from an associated data source (typically a data provider) and generate therefrom a displayable graphical representation of the received data according to an operative graphical data display mechanism. The data instances may be Document Object Model document instances."
>
> (Emphasis Added)

Both *Sandhu* and *Knight* use XSLT transformation for transforming a first XML document type into a second XML document type to achieve overall solution. However, the

claim invention is an alternative solution for XSLT. As indicated above, the claimed invention improves the performance of data transformation compared with XSLT transformation.

C. Claim 1

Claim 1 as amended recites:

> A method of data object transformation between a middleware and a application, the method comprising:
> receiving a message from a messaging middleware by a data transformation adapter, the message including one or more data objects of a first object type, wherein the message is a first communications format;
> converting by the data transformation adapter the message from the first communications format to a second communications format;
> ***converting by the data transformation adapter the one or more data objects from the first object type to a second object type***, wherein the one or more data objects are converted using a first set of one or more transformation classes, the one or more transformation classes being configured to transform the one or more data objects from the first object type to the second object type, each of the one or more transformation classes generated using mapping rules, ***the mapping rules including eXtensible Markup Language (XML) based syntax that uses rule specification guide to facilitate transforming the one or more data objects from the first object type to the second object type***; and
> transmitting by the data transformation adapter the one or more second object type data objects to an application.

(Emphasis Added)

The Final Office Action admits that "*Sandhu* does not explicitly teach wherein the one or more data objects are converted using a first set of one or more transformation classes, the one or more transformation classes being configured to transform the one or more data objects from the first object type to the second object type, each of the one or more transformation classes generated using mapping rules." The Final Office Action cited *Knight* to address the deficiency of *Sandhu*, stating that "*Knight* explicitly teaches generating transformation classes ("transformer object") based on retrieved XSLT documents containing mapping rules (para. [0012], [0081], [0101] and FIG. 13)." (page 4 of the Final Office Action).

Applicants respectfully disagree with the Final Office Action. *Knight* discloses a transformer object, but not for "transform[ing] the one or more data objects from the first object type to the second object type, each of the one or more transformation classes generated

using mapping rules", as recited in claim 1. In fact, *Knight* discloses a "transformer object 57 which may be used to transform the application-data-centric DOM ***XML document instance*** 54 to the presentation-centric DOM ***XML document instance*** 56." (para. 0081 of *Knight,* Emphasis Added).

The claimed invention focuses on transforming objects from one format to another by a transformation adapter, which is shown in Fig. 1 of the present application. The transformation adapter exposes application functionalities on a middleware, such as, RMI, TIBCO, CORBA, *etc*. ***More specifically, the claimed invention converts data objects are using transformation classes, which are configured to transform the data objects from the first object type to the second object type. The transformation classes are generated using mapping rules, which include eXtensible Markup Language (XML) based syntax that uses rule specification guide to facilitate transforming the one or more data objects from the first object type to the second object type***. The second object type is not in XML format which is the typical result of an XSLT transformation. In addition, the second object type can be used in application programming interface and does not involve run-time interpretation of transformation specification as in XSLT transformation.

Applicants respectfully assert that *Sandhu* in view of *Knight* focuses on a system that uses XSLT transformation to achieve overall solution. The XSLT transformation typical transforms a first XML document type into a second XML document type. Therefore, Applicants respectfully submit that *Sandhu* in view of *Knight* does not teach, disclose, or suggest ***transforming the data objects from the first object type to the second object type***, as recited in claim 1. Consequently, for at least this reason, among others, Applicants respectfully request that claim 1 be allowed and the rejection be withdrawn.

D. Independent claims 11, 17, 22 and 26

Applicants respectfully submit that for reasons related to those discussed above, *Sandhu* in view of *Knight* fails to teach, disclose, or suggest each and every element of independent claims 11, 17, 22 and 26.

E. Dependent Claims 2-6, 12-16, 18-21, and 23-25

Because independent claims 1, 11, 17, 22 and 26 are allowable over the cited art of record, dependent claims 2-6, 12-16, 18-21, and 23-25 are allowable as a matter of law for at least the reason that dependent claims 2-6, 12-16, 18-21, and 23-25 contain all features and elements of their respective independent base claim. *In re Fine*, 837 F.2d 1071, 5 U.S.P.Q.2d 1596, 1600 (Fed. Cir. 1988). Accordingly, the rejection to dependent claims 2-6, 12-16, 18-21, and 23-25 should be withdrawn for at least this reason, among others.

CONCLUSION

Applicants respectfully submit that the pending claims are in condition for allowance. Favorable reconsideration and allowance of the present application and all pending claims are hereby courteously requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned attorney at (770) 933-9500.

Respectfully submitted,

/Minh N. Nguyen/

Minh N. Nguyen
Registration No. 53,864